AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

LISTING OF CLAIMS

1. (Previously presented) A method for managing registers in a processor, comprising:

introducing a name level instruction for at least one of a named architected register into an instruction set, the instruction set for use with the processor;

allowing a programmer to change the current name level of the at least one named architected register via said name level instruction, the at least one named architected register available to the programmer as an additional named architected register as a result of a name level change; and

assigning a new physical memory location to the at least one named architected register upon receipt of the name level instruction instituting a name level change for the at least one named architected register.

2. (Previously presented) A method as in claim 1, wherein the physical memory location comprises a physical register, the name level instruction initiating hardware register renaming operations performed by hardware register renaming apparatus of the processor.

3. (Previously presented) A method as in claim 2, wherein the hardware register renaming apparatus maintains a pointer to a current physical register for the at least one named architected register, where upon receipt of the name level instruction the hardware register renaming apparatus changes the pointer to another physical register for the at least one named architected register with the new name level, thereby assigning a new physical memory location to the at least one named architected register.

4. (Previously presented) A method as in claim 1, wherein the processor uses at least one register stack to assign a new physical memory location to the at least one named architected register upon receipt of the name level instruction.

5. (Previously presented) A method as in claim 1, wherein the processor uses a hardware-managed register cache to assign a new physical memory location to the at least one named architected register upon receipt of the name level instruction.

6. (Previously presented) A method as in claim 1, wherein the processor uses a hardware-managed special-purpose memory to assign a new physical memory location to the at least one named architected register upon receipt of the name level instruction.

7. (Previously presented) A method as in claim 1, wherein the processor uses a hardware-managed component of main memory storage of a system to assign a new physical

memory location to the at least one named architected register upon receipt of the name level instruction.

8. (Previously presented) A method as in claim 1, wherein the processor uses a software-managed component of main memory storage of a system, where upon finding that a name resides in a special purpose area of main storage, an interrupt to the processor causes invocation of an interrupt handler that performs a task of bringing a value of the name level from the main storage to a physical register.

9. (Previously presented) A method as in claim 1, wherein the processor uses a hardware-managed hierarchy of structures such as cache and storage, successively larger in size and slower in access time to assign a new physical memory location to the at least one named architected register upon receipt of the name level instruction.

10. (Original) A method as in claim 1, wherein the name level instruction provides for the facilitation of architectural features which overload the architected register namespace reducing the overhead of register management.

11. (Previously presented) A method as in claim 1, wherein the name level instruction provides for additional architected registers without changing the instruction format of the computer.

12. (Currently amended) A computer program product comprising a computer ~~readable~~ usable memory medium ~~tangibly embodying~~ including a computer readable program, wherein the computer readable program ~~adapted~~ included in the computer usable memory medium causes a computer to perform operations when executed by a computer processor, the operations comprising:

receiving a name level instruction;

changing a current name level of a named architected register in response to the received name level instruction;

adding a new architected register for use by a programmer, the new architected register corresponding to the named architected register with the new name level instituted by the received name level instruction; and

assigning a new physical memory location to the at least one named architected register upon receipt of the name level instruction instituting a name level change for the named architected register.

13. (Previously presented) A computer program product as in claim 12, wherein the physical memory location comprises a physical register of the computer processor, the name level instruction initiating hardware register renaming operations performed by hardware register renaming apparatus of the computer processor.

14. (Original) A computer program product as in claim 12, wherein the name

level instruction provides for the facilitation of architectural features which overload the architected register namespace reducing the overhead of register management.

15. (Original) A computer program product as in claim 12, wherein the name level instruction provides for additional computer registers without changing the instruction format of the computer.

16. (Currently amended) A data processor comprising:

a memory ~~storing~~ configured to store a program comprising a plurality of instructions and at least one name level instruction, the name level instruction ~~changing~~ configured to change a name level of a named architected register when executed;

at least one execution unit ~~adapted~~ configured to execute the program comprising a plurality of instructions and at least one name level instruction;

said data processor further comprising a plurality of registers;[[,]] and

a register renaming mechanism coupled to a stack of register names and responsive to a name level instruction for assigning a new physical register to the named architected register upon execution of the at least one name level instruction of the program.

17. (Cancelled)

18. (Cancelled)

19. (Cancelled)

20. (Original) A data processor as in claim 16, wherein the memory includes a backing store.

21. (Previously presented) A data processor as in claim 16, wherein the register renaming mechanism used for hardware register renaming maintains a pointer to a current physical register for a corresponding architected register.

22. (Original) A data processor as in claim 16, wherein the name level instruction provides for creating additional computer registers without changing the instruction format of the computer.

23. (Previously presented) A method comprising:

receiving a name level instruction during execution of a program in a processor, the name level instruction changing a name level of a named architected register from an original name level to a new name level, the named architected register with the new name level treated as an additional named architected

register;

assigning a new physical memory location to the named architected register upon receipt of the name level instruction, the selection of which new physical memory location to be assigned to the named architected register with the new name level controlled solely by apparatus of the processor;

receiving additional instructions using the named architected register with the new name level;

using the new physical memory location assigned to the named architected register with the new name level during execution of the additional instructions; and

receiving a name level instruction changing the name level of the named architected register back to the original name level, removing the additional named architected register from availability for use.

24. (Previously presented) The method of claim 22 where the physical memory location is an internal register selected by register renaming apparatus of the processor.

25. (Previously presented) The method of claim 22 where the physical memory location is a general memory location of a system.